

SHIFT REGISTER IMPLEMENTATIONS
OF FIRST-IN/FIRST-OUT MEMORIES

This application claims the benefit of provisional patent application No. 60/200,644, filed 5 April 28, 2000, which is hereby incorporated by reference herein in its entirety.

Background of the Invention

This invention relates to first-in/first-out ("FIFO") memories, and more particularly to 10 implementing such memories using shift registers, for example, in a programmable logic device.

FIFO memories can be used to help transfer data between two different clock regimes that may not be synchronized with one another. For example, data 15 may be received in synchronism with a so-called "write clock" and may require further processing in the order received but in synchronism with a so-called "read clock" which is different from and perhaps not even synchronized with the write clock. A FIFO memory may 20 therefore be used as a data buffer between the circuitry associated with the write clock and the circuitry associated with the read clock. The FIFO memory takes in data in synchronism with the write clock, and the currently oldest data is read out in 25 synchronism with the read clock.

Generally, a FIFO memory must be equipped with circuitry to detect full and empty conditions of the memory. When the memory is full, writing must be stopped (e.g., to prevent loss of data). When the 5 memory is empty, reading must be stopped (e.g., to prevent the reading circuitry from erroneously operating on null data). Full and empty conditions can be detected by counting read and write clock pulses and comparing (e.g., subtracting) those counts. If the 10 write clock pulse count exceeds the read clock pulse count by an amount equal to the capacity of the FIFO memory, the memory is full and further writing should be stopped. If the read clock pulse count is equal to the write clock pulse count, the FIFO memory is empty 15 and further reading should be stopped. Of course writing or reading can be resumed as soon as the FIFO memory is no longer full or empty.

When the read and write clocks are not synchronized with one another, the counts of read and 20 write clock pulses are also not synchronized with one another and it can be difficult to reliably detect full and empty conditions by comparing those counts.

Summary of the Invention

In accordance with the present invention Gray 25 code counter circuitries are used to count both the read and write clocks. Gray code subtractor circuitry is used to process the Gray code outputs of the counters to produce a Gray code output which is used to address shift register circuitry for reading. Data is 30 shifted into the shift register circuitry in synchronism with the write clock and is read from the shift register circuitry location identified by the Gray code output of the Gray code subtractor. Certain

Gray code subtractor outputs indicate that the shift register circuitry is either full or empty. By using Gray code in this manner, a FIFO memory can be provided which has reliable and glitch-free full, empty, and 5 data output signals. The circuitry can be readily implemented in appropriately constructed programmable logic devices ("PLDs"), e.g., PLDs that include Gray code shift register capabilities.

Further features of the invention, its nature 10 and various advantages, will be more apparent from the accompanying drawings and the following detailed description.

Brief Description of the Drawings

FIG. 1 is a simplified schematic block 15 diagram of an illustrative FIFO memory constructed in accordance with the invention.

FIG. 2 is a table that is useful in explaining certain aspects of the construction and operation of the FIG. 1 circuitry in accordance with 20 the invention.

FIG. 3 is a more detailed but still simplified block diagram of an illustrative embodiment of a portion of the FIG. 1 circuitry in accordance with the invention.

FIG. 4 is another more detailed but still simplified block diagram of an illustrative embodiment of a portion of the FIG. 1 circuitry in accordance with the invention.

FIG. 5 is a more detailed but still simplified block diagram of an illustrative embodiment of another portion of the FIG. 1 circuitry in accordance with the invention.

FIG. 6 is a chart that is useful in explaining certain aspects of the operation of the circuitry of the invention.

FIG. 7 is a simplified block diagram of an 5 illustrative system employing circuitry in accordance with the invention.

Detailed Description

An illustrative FIFO memory 10 constructed in accordance with this invention is shown in FIG. 1.

10 FIFO memory 10 receives a write clock signal ("WCLK") via lead 12 and a read clock signal ("RCLK") via lead 14. The WCLK and RCLK signals do not need to be synchronized with one another in any respect. In the embodiment being discussed, however, it is assumed that 15 after any "clear" control signal is deasserted, the first edge of a "clock" signal is a rising edge. It is also assumed that a "clear" control signal clears components in FIG. 1 to the following values: circuitry 20 is cleared to 0001, circuitry 30 is 20 cleared to 0001, circuitry 40 is cleared to zero, and circuitry 60 is cleared to zero. (These assumptions are only illustrative, and other assumptions and combinations of operating parameters can be used in other implementations that are within the scope of this 25 invention.)

In addition to the WCLK and RCLK signals, FIFO memory 10 receives eight bits of parallel data ("FIFO WRITE DATA") via eight leads 16, and outputs eight bits of parallel data ("FIFO READ OUTPUT") via 30 eight leads 70. Successive eight-bit words of FIFO WRITE DATA are applied to FIFO memory 10 in synchronism with successive WCLK signal pulses. The oldest word of data in the FIFO memory 10 that was not read during a

previous RCLK signal cycle is read out via leads 70 during each successive RCLK signal cycle. FIFO memory 10 has the capacity to store as many as eight, eight-bit data words. These word length and memory capacity 5 parameters are only illustrative, and it will be readily apparent from the following discussion how these parameters can be increased or decreased if desired.

In FIG. 1 the elements indicated by reference 10 number 20 constitute Gray code counter circuitry for counting successive WCLK signal pulses and producing four output signals on leads 24 that are indicative of the current count in a Gray code. As is well known, a Gray code is one in which only one binary digit ("bit") 15 or place changes value when moving from any word to the next higher or lower word in the sequence of Gray code words. For example, FIG. 2 shows an example of a four-bit Gray code accompanied by a decimal equivalent value for each Gray code word. Each of the 16 Gray code 20 words is unique. The most significant bit ("MSB") "D" of each Gray code word is shown on the left, and the least significant bit ("LSB") "A" is shown toward the right. In each Gray code word the single bit that changes when moving from the preceding (lower decimal 25 valued) word is indicated by an asterisk. (This includes moving from the bottom-most word in FIG. 2 to the top-most word.) As has been said, the rule that only one bit changes in value when moving to the adjacent word is equally applicable when moving up the 30 Gray code sequence. The particular Gray code shown in FIG. 2 is only illustrative. There are other known Gray codes, and any Gray code can be used in connection with this invention. As is well known, a Gray code has the advantage that the state of only one bit position

is momentarily uncertain when going from one Gray code word to the next successive Gray code word (forward or backward).

Returning to FIG. 1, Gray code counter 20 can

5 be implemented in any of a variety of ways (e.g., as a
finite state machine outputting the desired Gray code
sequence in response to successive WCLK signal cycles).
However counter 20 is implemented, the output signals
should be free of "glitches" (i.e., momentarily
erroneous or indefinite states). FIG. 1 shows an
embodiment that may be particularly desirable when FIFO
memory 10 is implemented on an appropriately
constructed programmable logic device. (An example of
a programmable logic device suitable for use in
15 implementing this and all other portions of FIFO memory
10 is shown in commonly assigned, concurrently filed
U.S. patent application No. _____ (Docket No. 09/1761602
174/189), which is hereby incorporated by reference
herein in its entirety. Programmable logic device
20 implementation of FIFO memory 10 is further discussed
later in this specification.) In the embodiment shown
in FIG. 1 four logic modules 22 on the programmable
logic device are programmed to count successive WCLK
signal pulses and to produce a Gray code output 24
25 indicative of the count. In particular, each of logic
modules 22 produces a respective one of the four-bits
of the successive Gray code output words.

Counter 20 is preferably configured so that it successively outputs two successive Gray code values in response to each WCLK signal cycle. Considering again the illustrative Gray code shown in FIG. 2, it will be noted that the A or LSB place of that code alternates between logic 1 for two successive words and then logic 0 for the next two successive words.

Moreover, the changes in state of the A or LSB bit are all interleaved between changes in state in the other bits (B-D) considered collectively. Accordingly, the Gray code shown in FIG. 2 can be readily produced by

5 having the A or LSB bit clocked by the inverse of the WCLK signal, while the other bits (B-D) are clocked by and appropriately count pulses in the uninverted WCLK signal. Consistent with this description, counter 20 is shown in FIG. 1 receiving both the true and

10 complement of the WCLK signal. (In particular, as indicated by the "0" in FIG. 1, the logic module 22 producing the LSB receives the inverted WCLK signal, and as indicated by the "1-3" in FIG. 1, the three logic modules 22 producing the more significant bits

15 receive the true of the WCLK signal.) In effect, counter 20 is counting at twice the WCLK rate. Because the maximum capacity of FIFO memory 10 is eight words, and because counter 20 is counting at twice the WCLK rate, counter 20 is a recirculating counter having 16

20 different Gray code output states (e.g., the 16-word Gray code sequence shown in FIG. 2, with the bottom-most state being succeeded by return to the top-most state). (It will be remembered that whenever counter 20 is cleared, it clears to 0001.)

25 Counter 30 is another counter based on the same Gray code as counter 20. Counter 30 counts RCLK signal pulses. Unlike counter 20, however, counter 30 preferably counts only at the basic RCLK rate, not at twice that rate. Also unlike counter 20, the outputs

30 of counter 30 increment by two Gray code increments in response to each successive RCLK cycle. The intervening Gray code words are skipped. The output signals of counter 30 are therefore not a true Gray code, but they are based on a Gray code, (in particular

the Gray code implemented by counter 20). Thus, for example, if counter 20 uses the Gray code shown in FIG. 2, counter 30 may successively output the words in FIG. 2 having equivalent decimal values 1, 3, 5, 7, 9, 5 11, 13, 15, 1, etc., in response to successive RCLK signal pulses. For convenience herein the type of output sequence produced by counter 30 may sometimes be referred to as a "double-increment Gray code." (Again it will be remembered that whenever cleared, counter 30 10 clears to 0001.)

Counter 30 may be implemented in the same way that counter 20 is implemented. Like the outputs of counter 20, the outputs of counter 30 should be glitch-free. In the programmable logic device embodiment 15 shown in FIG. 1, each of four appropriately programmed logic modules 32 produce a respective one of the double-increment Gray code bits on a respective one of leads 34.

The Gray code output signals of counter 20 20 are shifted into shift register circuitry 40 in synchronism with the WCLK signal. Shift register circuitry 40 preferably comprises four shift register subcircuits 42, one for each of the four outputs 24 of counter 20. Each shift register subcircuit 42 25 preferably includes a series of 16 latches (suggested in part by the dotted lines toward the left in each subcircuit 42 in FIG. 1). The latches in each of these series may sometimes be referred to as eight master latches interspersed with eight slave latches. The 30 master latches receive and store data from the associated lead 24 or the preceding slave latch during one half of each WCLK signal cycle, and the slave latches receive and store data from the preceding master latch during the other half of each WCLK signal

cycle. (In the example in which the LSB portion of counter 20 is clocked by the inverse of WCLK and the other portions of counter 20 are clocked by WCLK, it is desirable to clock the LSB portion of shift register 40 by WCLK and to clock the other portions of circuitry 40 by the inverse of WCLK. Consistent with this description, FIG. 1 shows circuitry receiving both the true and complement of WCLK, with the "0" logic module receiving the true WCLK signal and the "1-3" logic modules receiving the complement WCLK signal.) From the foregoing it will be seen that as counter 20 produces successive Gray code words, those words are taken into and shifted down through shift register circuitry 40.

15 In addition to the above-described ability to shift in and down signals received via leads 24, shift register circuitry 40 includes selection or decoding circuitry for selecting any master or slave latches and outputting via leads 50 the contents of the selected latches. This selection circuitry makes its selections based on the outputs of counter 30 received via leads 34.

25 FIG. 3 shows how a representative one of shift register subcircuits 42 may be constructed in the case in which the Gray code shown in FIG. 2 is used. As shown in FIG. 3, representative shift register subcircuit 42 includes shift register components 44 and selection circuitry components 46. Components 42 include the above-described series of eight master latches interspersed with eight slave latches. Data supplied via the associated lead 24 to the DIN input terminal is shifted down through these latches in synchronism with successive half cycles of the CLK

signal. All of the latches may be cleared in parallel by asserting the CLR signal.

Selection circuitry 46 is a binary decoding tree for selecting one of inputs C0-C15 and basing 5 output signal 50 on the selected input. The selection is based on a binary decode of the four inputs D-A from leads 34. For example, when inputs D-A have values 0000, input C0 is selected; when inputs D-A are 0001, input C1 is selected; when inputs D-A are 0010, input 10 C2 is selected; and so on. In parentheses next to each input C0-C15 are the values of inputs D-A that cause selection of that input C0-C15.

The master and slave latch outputs MASTER 0 through SLAVE 7 are respectively connected to inputs 15 C0-C15 in an order based on the same Gray code as is used by counter 20. The illustrative order of the connections shown in FIG. 3 is based on the Gray code shown in FIG. 2. In particular, this Gray code order of the FIG. 3 connections is such that successive D-A 20 input words from leads 34 (also based on this Gray code) cause latches in circuitry 44 to be selected in the order that they appear in the shift register series. Bearing in mind that counter 30 outputs double-increment Gray code signals, the D-A signals may 25 at some illustrative time change from 0001 to 0010 (i.e., from decimal equivalent value 1 in FIG. 2 to decimal equivalent value 3) in response to counting an RCLK pulse. This will cause circuitry 46 in FIG. 3 to switch from selecting the SLAVE 0 output of circuitry 30 44 to selecting the SLAVE 1 output of circuitry 44. The next RCLK pulse will cause signals D-A to switch from 0010 to 0111 (decimal value 5 in FIG. 2). This will cause circuitry 46 in FIG. 3 to switch from selecting the SLAVE 1 output of circuitry 44 to

selecting the SLAVE 2 output of circuitry 44. Thus successive increases in the RCLK count output by counter 30 cause successive SLAVE latch outputs to be selected by circuitry 46 (looping back to SLAVE 0 after 5 SLAVE 7 has been selected).

In general (and for the moment ignoring details that will be considered later), the contents of the slave latches in circuitry 44 are historical values of the WCLK count signals output by counter 20 in Gray 10 code. A recent count value is stored in the lowest numbered slave latch (SLAVE 0), and earlier count values are stored in successively higher numbered slave latches. Assume for the moment that five WCLK signal pulses have been counted by counter 20 before the first 15 RCLK signal pulse is counted by counter 30. The SLAVE 5 latches of circuitry 40 will be storing a Gray code value (0001) indicative of the state to which counter 20 is cleared, the SLAVE 4 latches will be storing a Gray code count (0010) indicative of one complete WCLK 20 cycle (see FIG. 2 and remember that counter 20 counts at twice the WCLK rate), the SLAVE 3 latches will be storing a Gray code count (0111) indicative of two complete WCLK cycles, the SLAVE 2 latches will be storing 0100 indicative of three complete WCLK cycles, 25 the SLAVE 1 latches will be storing 1101 indicative of four complete WCLK cycles, and the SLAVE 0 latches will be storing 1110 indicative of five complete WCLK cycles.

Prior to counting the first RCLK signal 30 pulse, counter 30 outputs signals (0001) via leads 34 that cause circuitries 46 to select the SLAVE 0 outputs of the associated circuitries 44. Accordingly, shift register circuitry 40 will output (via leads 50) signals 1110. This indicates that WCLK is five cycles

ahead of RCLK. Signals 50 are applied to full/empty detector circuitry 80, which is configured to detect signals 50 having values 0000 or 0001 as indicative of an empty condition, and to detect signals valued 1001 5 or 1000 as indicative of a full condition. Because signals 50 having values 1110 indicate neither of these memory capacity extremes, circuitry 80 produces neither a FULL output signal on lead 82 nor an EMPTY output signal on lead 84. Accordingly, both reading and 10 writing are permitted to continue.

When the first RCLK pulse is counted by counter 30, circuitry 40 responds by switching to output the contents of the SLAVE 1 latches. In the example being discussed, this means that circuitry 40 15 now outputs 1101 via leads 50. This indicates that the WCLK count still exceeds the RCLK count by four, and so full/empty detector 80 still continues to permit reading and writing by outputting neither a FULL nor an EMPTY signal.

If the RCLK count eventually reaches five, circuitry 40 will output the contents of its SLAVE 5 latches (i.e., 0001 in the example being discussed). This will cause circuitry 80 to detect that reading has caught up with writing and that the FIFO is empty. 25 Circuitry 80 produces an EMPTY output signal, which is used to inhibit further reading until after at least one further write operation (after which the EMPTY signal is again turned off).

If during the course of the example being 30 discussed, further write operations occur, the WCLK values in the successive slave latches will increase, tending to widen the difference between the larger WCLK count read out from the slave latches selected by the smaller RCLK count. However, as soon as this

difference becomes eight, circuitry 40 will output 1000. Circuitry 80 will recognize this as a full condition of the FIFO and will output a FULL signal to stop further write operations until at least one 5 further read operation has relieved the FULL condition.

From the foregoing it will be seen that as counter 20 continues to count, all count values effectively recirculate through circuitries 44. These include values 0000 and 0001 (used by circuitry 80 to 10 detect an empty condition) and values 1001 and 1000 (used by circuitry 80 to detect a full condition). As counter 30 also continues to count, the slave latch selection controlled by that count also recirculates (from the top slave latch (SLAVE 0), progressively down 15 to the bottom slave latch (SLAVE 7), back to the top slave latch, and then progressively down to the bottom slave latch again). If reading is going too slowly relative to writing, the more rapid recirculation of the write count will cause the read count to eventually 20 point to (i.e., select) the slave latches containing the values (e.g., 1000) that indicate a full condition. This will cause circuitry 80 to output a FULL signal. On the other hand, if writing is going too slowly relative to reading, the more rapid recirculation of 25 the read count will cause the read count to eventually point to (i.e., select) the slave latches containing the values (e.g., 0001) that indicate an empty condition. This will cause circuitry 80 to output an EMPTY signal.

30 From the foregoing, it will be seen that circuitry 40 is effectively a subtractor operating on Gray code or Gray-code-based inputs. In particular, circuitry 40 effectively subtracts the Gray-code-based inputs on leads 34 from the Gray code inputs on leads

24. Circuitry 40 produces Gray code output signals indicative of the difference between these two groups of inputs, including outputs that indicate when the difference is either zero or an acceptable maximum. In 5 view of its thus-described capabilities, circuitry 40 may be sometimes referred to as a Gray code subtractor.

The circuitry described above has the advantage of always being able to detect either a full or empty condition of the FIFO, even though the WCLK 10 signal and the RCLK signal are not synchronized with one another. This is so because (1) counter 20 counts at twice the WCLK rate, (2) shift register circuitry 40 also shifts the output signals of counter 20 at twice the WCLK rate, and (3) there are two adjacent Gray code 15 WCLK counts that are recognized by circuitry 80 as indicating a full condition, and two more adjacent Gray code WCLK counts that are recognized by circuitry 80 as indicating an empty condition. FIG. 4 can be used to help further illuminate these points.

20 FIG. 4 shows the shift register 40 portion of FIG. 1 in more detail. Each of subcircuits 42 in FIG. 4 includes the features shown in FIG. 3. FIG. 4 shows that the shift register portion 44 of each subcircuit includes a gate ("G.") before the first 25 master latch ("M.L. 0") and between each master latch ("M.L.") or slave latch ("S.L.") and the succeeding master or slave latch. In each subcircuit 42 the gates G are divided into two interleaved groups. One group is controlled by the WCLK signal, and the other group 30 is controlled by the WCLK bar (WCLK inverse) signal. The polarity of the application of the WCLK signal is reversed for the right-most subcircuit 42, which handles the LSB signal A from counter 20. Moving toward the left in FIG. 4, each succeeding subcircuit

42 handles the next more significant bit of the outputs of counter 20. The selection control circuitry 46 of each subcircuit 42 receives and is controlled by the four outputs of counter 30 on leads 34.

5 To provide a brief, illustrative example of the operation of the FIG. 4 circuitry, when WCLK goes high, the top-most gate G in the left-most subcircuit 42 is enabled. This allows M.L. 0 in that subcircuit to receive the signal on the associated lead 24 from 10 counter 20. When the WCLK signal goes low, M.L. 0 latches in that signal from lead 24 and begins to apply that signal to S.L. 0 through the intervening gate G, which is now enabled by the WCLK bar signal. When WCLK goes high again, S.L. 0 latches in the signal it 15 received from M.L. 0 and begins to apply that signal to M.L. 1 through the intervening gate G, which is now enabled by the WCLK signal. The top-most gate G is also enabled again so that M.L. 0 can receive the next counter 20 output value from the associated lead 24.

20 In this way data shifts into and down through the shift register components 44 of each of the three left-most subcircuits 42 in synchronism with the WCLK bar signal and of the right-most subcircuit 42 in synchronism with the WCLK signal.

25 To facilitate further discussion, in FIG. 4 each master and slave stage box is shown containing the data that it contains at the start of the example discussed above in connection with FIG. 3. Down through S.L. 5 the data shown in FIG. 4 was output by 30 counter 20 in response to five WCLK signal cycles. Below S.L. 5 the data is all zeroes because the example being considered assumes that there was a clear operation, and circuitry 40 is cleared to zero as mentioned earlier in this specification.

Suppose now that just before the condition shown in FIG. 4 is reached, an RCLK pulse occurs which causes all the selection circuits 46 in FIG. 4 to switch from selecting the slave 4 latches to selecting the slave 5 latches. Suppose also that at about the same time a transition is occurring in the WCLK signal. (This is the WCLK transition that will actually result in the data pattern shown in FIG. 4.) If the RCLK pulse occurs before the WCLK transition, the slave 5 latches will still contain 0000 when the RCLK pulse occurs. If the RCLK pulse occurs after the WCLK transition, the slave 5 latches will contain 0001 when the RCLK pulse occurs. Circuitry 80 (FIG. 1) recognizes either 0000 or 0001 as indicating an empty condition. Accordingly, circuitry 80 produces a stable and glitch-free EMPTY output signal even though WCLK and RCLK are not synchronized with one another and even though a WCLK signal transition may be occurring at about the same time as an RCLK signal pulse is occurring. (As a practical matter, shift register 44 executes a data shift more rapidly than selection circuitries 46 can change output state.) Thus an empty condition can always be reliably detected even though there is no synchronization between WCLK and RCLK. Moreover, the same principles apply to detection of a full condition, so that a full condition is also detected as reliably as an empty condition.

If the invention is being implemented in a programmable logic device, it can be particularly helpful to use programmable logic device circuitry of the type shown in above-mentioned application No.

(Docket No. 174/189). This is so because that reference shows programmable logic device logic modules that can perform all the functions

5/22
30

performed by subcircuits 42 herein. In particular, all the functions performed by one subcircuit 42 herein can be performed in one logic module in the last-mentioned reference. In addition, the last-mentioned reference 5 includes clock signal handling capabilities associated with each group or block of logic modules that facilitate including all the circuitry in FIG. 4 in one block of logic modules. One logic module in the block can receive a normal clock signal (like that needed by 10 the right-most subcircuit 42 in FIG. 4) while other logic modules receive an inverted version of that same clock signal. The circuitry shown in the last-mentioned reference also facilitates stringing together 15 the shift register capabilities of logic modules so that if shift registers longer than those employed in the examples described herein are needed, such longer shift registers can be readily provided.

Returning again to FIG. 1, the FIFO data storage portion of memory 10 is shift register 20 circuitry 60. Like shift register circuitry 40, circuitry 60 is shown as made up of a plurality of shift register subcircuits 62, each of which may be similar to any one of the subcircuits 42 of circuitry 40. Thus each of subcircuits 62 is preferably 25 constructed as shown for subcircuits 42 in FIGS. 3 and 4. Because circuitry 60 is designed to store and output eight-bit data words, circuitry 60 includes eight subcircuits 62. A representative one of subcircuits 62 is shown in more detail in FIG. 5.

30 Each of subcircuits 62 receives a respective one of the eight bits of the FIFO WRITE DATA via a respective one of leads 16. Similarly, each of subcircuits 62 outputs a respective one of the eight bits of the FIFO READ OUTPUT via a respective one of

leads 70. Circuitry 60 shifts in the FIFO WRITE DATA in synchronism with the WCLK signal. Circuitry 60 outputs FIFO READ OUTPUT from master or slave latches of subcircuits 62 that are currently being selected in 5 response to the selection control signals currently being output (via leads 50) by Gray code subtractor 40. Storing data in circuitry 60 is straightforward and does not require any significant further discussion. Reading data from circuitry 60 is somewhat more subtle 10 and will now be considered in more detail.

Because WCLK is not synchronized with RCLK, the output of Gray code subtractor 40 can be any of the 16 Gray code values output by counter 20. Even though counter 30 can only cause the contents of slave latches 15 in circuitry 40 to be read out via leads 50, all 16 Gray code outputs of counter 20 are eventually shifted through each set of slave latches in circuitry 40. Just as there are two adjacent Gray code values associated with each of the full and empty conditions, 20 there are two adjacent Gray code values associated with each FIFO WRITE DATA word (because counter 20 counts at twice the WCLK rate). Each FIFO WRITE DATA word will also be stored in at least one of two adjacent sets of latches (one master latch set and one slave latch set) 25 in circuitry 60 at any given time. (Indeed, for most of each WCLK cycle (and each WCLK bar cycle) each FIFO WRITE DATA word will be stored in two adjacent sets of latches in circuitry 60.) Moreover, shifting of the Gray code data in the shift register portions 44 of 30 circuitry 40 occurs in synchronism with shifting of data in the shift register portions 64 of circuitry 60.

As will be apparent from the foregoing discussion, when an RCLK pulse occurs, circuitry 40 switches to outputting the contents of the slave

latches in circuitry 40 that contain the oldest previously unread Gray code count from counter 20. This will be the location (in Gray code) of the oldest previously unread FIFO WRITE DATA word in circuitry 60.

5 If a WCLK transition occurs at about the time that a read event is taking place, both the Gray code count information in circuitry 40 and the FIFO WRITE DATA in circuitry 60 will shift by one latch. Thus circuitry 60 will continue to output the correct FIFO DATA OUTPUT

10 15 even though the WCLK and RCLK are not synchronized with one another.

FIG. 6 is a further illustration of the point just stated. FIG. 6 shows the shifting of successive FIFO WRITE DATA words D0, D1, D2, etc., into the master and slave latches of the shift register portions 64 of subcircuits 62 in circuitry 60 (FIGS. 1 and 5) in synchronism with the WCLK signal. For example, the first FIFO WRITE DATA word D0 is present in master latch 0 during the initial low half-cycle of WCLK.

20 When WCLK thereafter goes high, D0 is latched into master latch 0 and is also applied to slave latch 0. When WCLK subsequently goes low again, D0 is latched into slave latch 0 and applied to master latch 1. At the same time, the next FIFO WRITE DATA word D1 is

25 applied to master latch 0. Data intake and shifting continues in this manner in synchronism with further WCLK signal cycles.

Assuming (as shown toward the left in FIG. 6) that during the initial few of the depicted WCLK signal cycles the RCLK signal does not change state, then circuitry 60 will select and output data from the latch locations indicated by the shaded upper right-hand corners in FIG. 6. This is so because the output of counter 30 is constant, thereby pointing constantly to

one set of slave latches in circuitry 40, but the Gray code counter 20 count information in circuitry 40 is advancing in synchronism with WCLK. This means that the output signals 50 of circuitry 40 are also an advancing Gray code sequence synchronized with WCLK, which causes the selection circuitries 66 in circuitry 60 to select successive sets of latches in that circuitry at the same rate that the FIFO WRITE DATA is being shifted through those latches. Circuitry 60 therefore continuously outputs D0 during the initially constant portion of the RCLK signal trace shown in FIG. 6. (In this connection it is again noted that circuitry 60 can execute a shift of data from one latch to the next faster than the outputs of its selection circuitry components 66 can change state. This helps maintain a constant D0 output 70 in the example being described.)

Toward the right in FIG. 6 RCLK becomes logic 1, indicating that it is now desired to read the oldest previously unread FIFO WRITE DATA in circuitry 60. If, in a worst-case situation depicted in FIG. 6, this RCLK event occurs at about the same time as a WCLK event, the FIFO WRITE DATA may be shifting in circuitry 60 just when it is desired to read data from that circuitry. Because of the way the circuitry is constructed and operates, however, this will not matter. If the FIFO WRITE DATA shifts before the read pointer (on leads 50) increments, the D0 data will be read from the master 3 latches of circuitry 60 as indicated at "b" in FIG. 6. On the other hand, if the read pointer (on leads 50) has incremented when the FIFO WRITE DATA shifts, the same D0 data will be read from the slave 3 latches of circuitry 60 as indicated at "a" in FIG. 6. This further demonstrates the

ability of the circuitry of this invention to operate without synchronism of the WCLK and RCLK signals.

Once again, if the invention is being implemented in a programmable logic device, it can be particularly helpful to use programmable logic device circuitry of the type shown in above-mentioned application No. (Docket No. 174/189) because (as has been mentioned) that circuitry includes logic modules that can readily perform all the functions required of subcircuits 62 herein. In particular, each subcircuit 62 can be implemented in one logic module in the last-mentioned reference. In addition, the last-mentioned reference shows circuitry that facilitates stringing together the shift register capabilities of logic modules if longer shift registers are needed to provide deeper FIFO memories (i.e., FIFO memories with capacity for more data words).

FIG. 7 illustrates FIFO memory 10 in accordance with this invention in a larger system 502 such as a data processing system. FIG. 7 shows the circuitry 10 of this invention implemented in programmable logic device ("PLD") 400 (although it will be understood that circuitry 10 could be alternatively implemented in other ways such as in special-purpose circuitry). System 502 may include one or more of the following components: a processor 504; memory 506; I/O circuitry 508; and peripheral devices 510. These components are coupled together by a system bus 520 and are populated on a circuit board 530 which is contained in an end-user system 540.

System 502 can be used in a wide variety of applications, such as computer networking, data networking, instrumentation, video processing, digital signal processing, or any other application where the

advantage of using FIFO memory circuitry 10 of the type shown herein is desirable. Programmable logic device 400 can be used to perform a variety of different logic functions. For example, programmable logic device 400

5. can be configured as a processor or controller that works in cooperation with processor 504. Programmable logic device 400 may also be used as an arbiter for arbitrating access to a shared resource in system 502. In yet another example, programmable logic device 400

10 can be configured as an interface between processor 504 and one of the other components in system 502. It should be noted that system 502 is only exemplary, and that the true scope and spirit of the invention should be indicated by the following claims.

15 It will be understood that the foregoing is only illustrative of the principles of the invention and that various modifications can be made by those skilled in the art without departing from the scope and spirit of the invention. For example, different Gray

20 codes can be used if desired. Similarly, FIFO memories of different depths (i.e., data word capacities) and/or widths (i.e., data word lengths) can be constructed using the principles of the invention. Also, different clocking arrangements can be used, with different

25 preset states of counters 20 and 30.